IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Vijay Tewari et al.		
		Art Unit:	2128
Serial No.:	10/808,991		
		Examiner:	Ferris III, Fred O
Filed:	March 24, 2004		

For: USE OF A VIRTUAL MACHINE TO EMULATE A HARDWARE DEVICE

REPLY TO EX PARTE QUAYLE ACTION

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This reply is in response to the Ex Parte Quayle Action in the instant application that was mailed on February 4, 2008. It is believed that a four-month extension of time fee is required. Please consider this a request therefore. It is not believed that extensions of time are required beyond those that may otherwise be provided for in documents accompanying this paper. However, if additional extensions of time are necessary to prevent abandonment of this application, then such extensions of time are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required therefore are hereby authorized to be charged to Deposit Account No. 50-0221.

In this Action, the Examiner indicates that the complex nature of the subject matter examined and the quantity of references submitted in Information Disclosure Statements (IDSs) place an exceptionally unreasonable burden on the Patent Office. As a result, the Examiner has requested assistance from Applicants with the references submitted in the IDSs. Applicants agree with the Examiner's position, apologize for what happened and have attempted to comply with the Examiner's request. As such, fifteen (15) IDS references are identified below.

The pending claimed subject matter involves similar features of: configuring a device virtual machine (VM) to emulate a hardware device, wherein the device VM includes device emulation code used to emulate the hardware device; listing the device VM as an available hardware device during VM bootstrap initialization; and allowing one or more other VMs to use the device VM as the emulated hardware device. It is believed that the identified references do not teach or suggest the claimed invention, and in particular the features of listing the device VM as an available hardware device during VM bootstrap initialization and allowing one or more other VMs to use the device VM as the emulated hardware device. A benefit or improvement over the identified references is to provide a mechanism via a VM for emulating hardware devices that are used to perform specialized functions or to emulate hardware devices that are too expensive (see, e.g., instant application on page 5, paragraph [0007]). Each of the identified references is discussed next.

The Dr. Gideon Frieder paper entitled "The Architecture and Operational Characteristics of the VMX Host Machine," IEEE 1982, pp. 9-16, USA (hereinafter referred to as "Frieder") states that the VMX host machine is a hardware-firmware environment for implementation of actual computer systems with a favorable price performance ratio. Frieder further states that it presents the host framework architecture, together with all units which are used to build actual systems. In addition, Frieder states the advantages and future potential of the architecture is briefly discussed. Frieder states that the notion of memory directives within the frame-work of an active (or "intelligent") memory unit is introduced. Frieder states that this notion serves as one example of the possibilities that are built into the architecture of the VMX host. (See, e.g., Frieder, page 9).

Frieder does not appear to teach or suggest all of the features of the claimed invention, and in particular the features of listing the device VM as an available hardware device during VM bootstrap initialization and allowing one or more other VMs to use the device VM as the emulated hardware device. Frieder does appear to discuss virtual machine technology, which is the technology of the claimed invention.

U.S. Patent No. 4,347,565 (Issue Date: August 31, 1982; Title: Address Control System for Software Simulation; Inventors: Saburo Kaneda et al.) (hereinafter referred to as "Kaneda") states that an address control system is provided for software simulation in a virtual machine system having a virtual storage function. Kaneda further states that when a simulator program is simulating an instruction of a program to be simulated, an address translation of an operand address in the program to be simulated is achieved using a translation lookaside buffer, thereby greatly reducing the overhead for the address translation during the simulator program execution. (See, e.g., Kaneda, Abstract).

Kaneda does not appear to teach or suggest all of the features of the claimed invention, and in particular the features of listing the device VM as an available hardware device during VM bootstrap initialization and allowing one or more other VMs to use the device VM as the emulated hardware device. Kaneda does appear to discuss virtual machine technology, which is the technology of the claimed invention.

The P. H. Gum paper entitled "System/370 Extended Architecture: Facilities for Virtual Machines" IBM J. Res. Develop. Vol 27, No. 6, pp. 530-543, November 1983 (hereinafter referred to as "Gum") states that it describes the evolution of facilities for virtual machines on IBM System/370 computers, and the elements of a new architectural facility

designed for the virtual-machine environment. In addition, Gum states that assists that have been added to various System/370 models to support the use of virtual machines are summarized, and a general facility for this purpose which was introduced with the System/370 Extended Architecture (370-XA) is described. Gum states that a new instruction of the 370-XA architecture places the machine in a specific mode in which several special capabilities are enabled. According to Gum, these allow the machine to provide execution in the virtual-machine environment of most of the instructions (including many privileged instructions) and most of the facilities (such as dynamic address translation) of both the System/370 and the 370-XA architectures. (See, e.g., Gum, page 530).

Gum does not appear to teach or suggest all of the features of the claimed invention, and in particular the features of listing the device VM as an available hardware device during VM bootstrap initialization and allowing one or more other VMs to use the device VM as the emulated hardware device. Gum does appear to discuss virtual machine technology, which is the technology of the claimed invention.

U.S. Patent No. 4,975,836 (Issue Date: December 4, 1990; Title: Virtual Computer System; Inventors: Toshio Hirosawa et al.) (hereinafter referred to as "Hirosawa") states that a virtual machine system has a processor operating as a base machine which includes a plurality of groups of registers, such as data registers, address registers and instruction address registers which are selected according to identification data identifying individual virtual machines. Hirosawa further states that a respective group of registers is allotted to each virtual machine and to the virtual machine monitor in the system. Hirosawa states that a status register file is provided to effect selective access to a respective one of a plurality of status registers in the

system in accordance with a virtual machine number. Further, Hirosawa states that an interrupt processing portion is provided comparing the value of an interrupt mask level of the status word of a corresponding virtual machine with an externally generated interrupt level to control interrupt processing. (See, e.g., Hirosawa, Abstract).

Hirosawa does not appear to teach or suggest all of the features of the claimed invention, and in particular the features of listing the device VM as an available hardware device during VM bootstrap initialization and allowing one or more other VMs to use the device VM as the emulated hardware device. Hirosawa does appear to discuss virtual machine technology, which is the technology of the claimed invention.

U.S. Patent No. 5,187,802 (Issue Date: February 16, 1993; Title: Virtual Machine System with Vitual Machine Resetting Store Indicating That Virtual Machine Processed Interrupt Without Virtual Machine Control Program Intervention; Inventors: Taro Inoue et al.) (hereinafter referred to as "Inoue") states that in a virtual machine system in which a virtual machine directly executes operations by use of the hardware without an intervention from the virtual machine control program (VMCP), at an occurrence of an input/output interruption, the system sets to a storage an event that the input/output interruption has been accepted and reserved by the VMCP. Inoue further states that when the virtual machine processes interruption information by means of the hardware without an intervention of the VMCP, the virtual machine resets the state of the storage. According to Inoue, when the virtual machine is set to an interruptible state, control is passed to the VMCP. Inoue states that the VMCP tests to determine whether or not the virtual machine has reset the state of the storage, thereby judging an acceptability of the interruption. (See, e.g., Inoue, Abstract).

Inoue does not appear to teach or suggest all of the features of the claimed invention, and in particular the features of listing the device VM as an available hardware device during VM bootstrap initialization and allowing one or more other VMs to use the device VM as the emulated hardware device. Inoue does appear to discuss virtual machine technology, which is the technology of the claimed invention.

U.S. Patent No. 5,255,379 (Issue Date: October 19, 1993; Title: Method for Automatically Transitioning From V86 Mode to Protected Mode in a Computer System Using an Intel 80386 or 80486 Processor; Inventor: Michael D. Melo) (hereinafter referred to as "Melo") states that a method for transitioning an Intel processor from virtual 8086 (V86) mode to protected mode operation which detects when a virtual V86 processor attempts to transition to protected mode, stores all of the information concerning the virtual processor at the time of the attempt to transition to protected mode, remaps the memory allotted to the virtual processor to the memory space used in running a process in real mode, sets up a dummy stack to provide for operation during a transition to protected mode, moves a process for transitioning to real memory space, shifts all of the register values to values for real memory space, and finally reactivates the transition to protected mode is provided. (See, e.g., Melo, Abstract).

Melo does not appear to teach or suggest all of the features of the claimed invention, and in particular the features of listing the device VM as an available hardware device during VM bootstrap initialization and allowing one or more other VMs to use the device VM as the emulated hardware device. Melo does appear to discuss virtual machine technology, which is the technology of the claimed invention.

U.S. Patent No. 5,319,760 (Issue Date: June 7, 1994; Title: Translation Buffer for Virtual Machines with Address Space Match; Inventors: Andrew H. Mason et al.) (hereinafter referred to as "Mason") states that a central processing unit (CPU) executing a virtual memory management system employs a translation buffer for caching recently used page table entries is provided. According to Mason, when more than one process is executing on the CPU, the translation buffer is usually flushed when a context switch is made, even though some of the entries would still be valid for commonly-referenced memory areas. Mason states that an address space number feature is employed to allow entries to remain in the translation buffer for processes not currently executing, and the separate processes or the operating system can reuse entries in the translation buffer for such pages of memory that are commonly referenced. Mason states that to allow this, an "address space match" entry in the page table entry signals that the translation buffer content can be used when the address tag matches, even though the address space numbers do not necessarily match. Mason states that when executing virtual machines on this CPU, with a virtual machine monitor, the address space match feature is employed among processes of a virtual machine, but an additional entry is provided to disable the address space match feature for all address space numbers for the virtual machine monitor. (See, e.g., Mason, Abstract).

Mason does not appear to teach or suggest all of the features of the claimed inventionand in particular the features of listing the device VM as an available hardware device during VM bootstrap initialization and allowing one or more other VMs to use the device VM as the emulated hardware device. Mason does appear to discuss virtual machine technology, which is the technology of the claimed invention.

U.S. Patent No. 5,455,909 (Issue Date: October 3, 1995; Title: Microprocessor With Operation Capture Facility; Inventors: James S. Blomgren et al.) (hereinafter referred to as "Blomgren") states that a microprocessor with a special Operation Capture Facility (OCF) mechanism is provided which enables "faulting" whenever there is (a) a memory access request to any one of a specified plurality of blocks of memory (b) a request to access any one of a plurality of specified I-O ports or (c) any one of a specified plurality of interrupts is activated. According to Blomgren, this OCF mechanism includes a plurality of special registers which store either (a) an I-O port address, (b) a memory address or (c) an interrupt number. Blomgren further states that mask registers are provided which (1) mask bits in the special register, thereby providing the ability to fault on an entire block of I-O access requests or upon activation of any one of a block of interrupts and (2) indicate which type of interrupts should be faulted and to indicate whether I-O should be faulted on a byte, word or double word. (See, e.g., Blomgren, Abstract).

Blomgren does not appear to teach or suggest all of the features of the claimed invention, and in particular the features of listing the device VM as an available hardware device during VM bootstrap initialization and allowing one or more other VMs to use the device VM as the emulated hardware device. Blomgren does appear to discuss virtual machine technology, which is the technology of the claimed invention.

U.S. Patent No. 5,459,869 (Issue Date: October 17, 1995; Title: Method for Providing Protected Mode Services for Device Drivers and Other Resident Software; Inventor: Michael L. Spilo) (hereinafter referred to as "Spilo") states that a method and system for allowing protected mode device drivers and resident programs to load and execute from an MS/PC-DOS environment are provided. Spilo further states that a further method enabling such protected

mode programs in an Intel x86 environment to transition between host environments is provided. Spilo further states that a method for allowing for protected mode programs running in a DOS environment which can then transition and continue to function in a Windows environment and that an improved method of mode switching for such drivers is also provided. (See, e.g., Spilo, Abstract).

Spilo also appears to discuss the switching of V86 mode code to protected mode code at initialization. (See, e.g., Spilo, col. 4, lines 7-24).

Spilo does not appear to teach or suggest all of the features of the claimed invention, and in particular the features of listing the device VM as an available hardware device during VM bootstrap initialization and allowing one or more other VMs to use the device VM as the emulated hardware device. Spilo does appear to discuss virtual machine technology, which is the technology of the claimed invention.

U.S. Patent No. 5,717,903 (Issue Date: February 10, 1998; Title: Method and Apparatus for Emulating a Peripheral Device to Allow Device Driver Development Before Availability of the Peripheral Device; Inventor: Thomas J. Bonola) (hereinafter referred to as "Bonola") states that a method of emulating a peripheral device in a multiprocessor computer system to test device driver programs is provided. According to Bonola, the emulation program is loaded by a host microprocessor into one or more of the other microprocessors (target microprocessors) which are not being accessed by the operating system software. Bonola further states that after the emulation program is loaded, control vectors to the entry point of the emulation program, where the environment in each of the target microprocessors are initialized for the emulator program. Bonola further states that if more than one target microprocessor are

utilized, then one of the target microprocessors are designated as the "master" microprocessor, which accepts interprocessor interrupts from the host microprocessor. According to Bonola, when the device driver program running on the host microprocessor invokes an I/O command, and emulation mode is selected, then an interprocessor interrupt (IPI) is asserted to the master microprocessor. Bonola states that, in response, an I/O emulation interrupt handler is executed by the master microprocessor to provide the appropriate responses to the device driver under test. (See, e.g., Bonola, Abstract).

Bonola further states that during the power up initialization phase of the device drivers, the emulation program is loaded by a host microprocessor from the hard disk drive to an allocated region in system memory. According to Bonola, after the emulation program is loaded, control vectors to the entry point of the emulation program. (See, e.g., Bonola, Summary of the Invention).

Bonola further states that numerous benefits result from having the emulation of the peripheral device executing on a separate or multiple separate processors from the host processor executing the device driver. Bonola states that first, program and task continuity are better maintained on the host processor, as large amounts of time are not spent running emulation code. Bonola states that the effective task flow on the host processor is very close to that which will occur in the final system and that the task flow change between the IPI communication and the actual I/O or memory operations is very minor when compared to the change required to run the emulator. According to Bonola, the basic operation of the host computer is much closer to final, allowing more complete development of the device driver, particularly the interaction with other tasks. Bonola states that second, as the separate processor

is by definition a bus master, and most advanced peripheral device controllers are bus masters, the interaction between the host processor and the peripheral device bus master is also better emulated. According to Bonola, it is easier to determine and develop the interface and communication than if the emulator code is running on the host processor. According to Bonola, improved bus master emulation is also provided, which also improves the time of device driver development and the use of the separate processor or processors better emulates the operation of a peripheral device under development than if the emulation code were run on the host processor. According to Bonola, this allows more complete device driver development prior to availability of the actual peripheral device, which in turn allows the peripheral device to be released sooner than otherwise reasonably possible, allowing a further increase in the hectic pace of technology improvement. (See, e.g., Bonola, col. 17, lines 19-48).

Bonola does not appear to teach or suggest all of the features of the claimed invention, and in particular the features of listing the device VM as an available hardware device during VM bootstrap initialization and allowing one or more other VMs to use the device VM as the emulated hardware device. Bonola does appear to discuss virtual machine technology, which is the technology of the claimed invention.

International Application WO 99/18511 (Publication Date: 04-15-1999; Title: Method and Apparatus for Providing Execution of System Management Mode Services in Virtual Mode; Inventor: David S. Edrich) (hereinafter referred to as "Edrich") states that the invention is an apparatus and method for executing instructions in a system management mode in a processor-based system. Edrich further states that the apparatus comprises a memory for storing instruction sequences by which the processor-based system is processed where the

memory includes a system management random access memory (SMRAM). Edrich states that the apparatus also comprises a processor having a system address space, that executes the stored instruction sequences. In addition, Edrich states that the stored instruction sequences include process steps to cause the processor to: (a) configure the processor to operate in a protected mode while in a system management mode, the processor operating at address greater than one megabyte; (b) invoke a paging feature of the processor; (c) configure the processor to operate in a virtual mode; and (d) process the instruction sequences; wherein the process steps occur upon the receipt of an instruction to process a system management request. (See, e.g., Edrich, Abstract).

Edrich does not appear to teach or suggest all of the features of the claimed invention, and in particular the features of listing the device VM as an available hardware device during VM bootstrap initialization and allowing one or more other VMs to use the device VM as the emulated hardware device. Edrich does appear to discuss virtual machine technology, which is the technology of the claimed invention.

U.S. Patent No. 6,035,374 (Issue Date: March 7, 2000; Title: Method of Executing Coded Instructions in a Multiprocessor Having Shared Execution Resources Including Active, Nap, and Sleep States in Accordance with Cache Miss Latency; Inventors: Ramesh Panwar et al.) (hereinafter referred to as "Panwar") states that a method of executing coded instructions in a dynamically configurable multiprocessor having shared execution resources including steps of placing a first processor in an active state upon booting of the multiprocessor is provided. According to Panwar, in response to a processor create command, a second processor is placed in an active state. Panwar further states that when either the first or second processor encounter a cache miss that has to be serviced by off-chip cache the processor requiring service is placed

in nap state in which instruction fetching for that processor is disabled. Panwar further states that when either the first or second processor encounter a cache miss that has to be serviced by main memory, the processor requiring services I placed in a sleep state by flushing all instructions from the processor in the sleep state and disabling instruction fetching for the processor in the sleep state. (See, e.g., Panwar, Abstract).

Panwar does not appear to teach or suggest all of the features of the claimed invention, and in particular the features of listing the device VM as an available hardware device during VM bootstrap initialization and allowing one or more other VMs to use the device VM as the emulated hardware device. Panwar does appear to discuss virtual machine technology, which is the technology of the claimed invention.

U.S. Patent No. 6,075,938 (Issue Date: June 13, 2000; Title: Virtual Machine Monitors for Scalable Multiprocessors; Inventors: Edouard Bugnion et al.) (hereinafter referred to as "Bugnion") states that the problem of extending modern operating systems to run efficiently on large-scale shared memory multiprocessors without a large implementation effort is solved by a unique type of virtual machine monitor. According to Bugnion, virtual machines are used to run multiple commodity operating systems on a scalable multiprocessor. Bugnion states that to reduce the memory overheads associated with running multiple operating systems, virtual machines transparently share major data structures such as the operating system code and the file system buffer cache. Bugnion states that we use the distributed system support of modern operating systems to export a partial single system image to the users. According to Bugnion, two techniques, copy-on-write disks and the use of a special network device, enable transparent resource sharing without requiring the cooperation of the operating systems. Bugnion states

that this solution addresses many of the challenges facing the system software for these machines and the overheads of the monitor are small and the approach provides scalability as well as the ability to deal with the non-uniform memory access time of these systems. According to Bugnion, the overall solution achieves most of the benefits of operating systems customized for scalable multiprocessors yet it can be achieved with a significantly smaller implementation effort. (See, e.g., Bugnion, Abstract).

Bugnion does not appear to teach or suggest all of the features of the claimed invention, and in particular the features of listing the device VM as an available hardware device during VM bootstrap initialization and allowing one or more other VMs to use the device VM as the emulated hardware device. Bugnion does appear to discuss virtual machine technology, which is the technology of the claimed invention.

U.S. Patent No. 6,397,242 (Issue Date: May 28, 2002; Title: Virtualization System Including a Virtual Machine Monitor for a Computer with a Segmented Architecture; Inventors: Scott W. Devine et al.) (hereinafter referred to as "Devine") states that in a computer that has hardware processor, and a memory, the invention provides a virtual machine monitor (VMM) and a virtual machine (VM) that has at least one virtual processor and is operatively connected to the VMM for running a sequence of VM instructions, which are either directly executable or non-directly executable. Devine further states that the VMM includes both a binary translation sub-system and a direct execution sub-system, as well as a sub-system that determines if VM instructions must be executed using binary translation, or if they can be executed using direct execution. Devine further states that shadow descriptor tables in the VMM, corresponding to VM descriptor tables, segment tracking and memory tracing are used as factors in the decision of which execution mode to activate. According to Devine, the

invention is particularly well-adapted for virtualizing computers in which the hardware processor has an Intel x86 architecture. (See, e.g., Devine, Abstract).

Devine also states that machine simulators, also known as machine emulators, run as application programs on top of an existing operating system. According to Devine, they emulate all the components of a given computer system with enough accuracy to run an operating system and its applications. Devine further states that machine simulators are often used in research to study the performance of multiprocessors. (See, e.g., Devine, col. 2, lines 37-43).

Devine does not appear to teach or suggest all of the features of the claimed invention, and in particular the features of listing the device VM as an available hardware device during VM bootstrap initialization and allowing one or more other VMs to use the device VM as the emulated hardware device. Devine does appear to discuss virtual machine technology, which is the technology of the claimed invention.

U.S. Patent Application No. 20040117539 A1 (Publication Date: June 17, 2004; Title: Methods and Systems to Control Virtual Machines; Inventors: Steve M. Bennett et al.) (hereinafter referred to as "Bennett") states that methods and systems are provided to control the execution of a virtual machine (VM). Bennett further states that a VM Monitor (VMM) accesses VM Control Structures (VMCS) indirectly through access instructions passed to a processor. According to Bennett, the access instructions include VMCS component identifiers used by the processor to determine the appropriate storage location for the VMCS components. The processor identifies the appropriate storage location for the VMCS component within the processor storage or within memory. (See, e.g., Bennett, Abstract).

Bennett does not appear to teach or suggest all of the features of the claimed invention, and in particular the features of listing the device VM as an available hardware device during VM bootstrap initialization and allowing one or more other VMs to use the device VM as the emulated hardware device. Bennett does appear to discuss virtual machine technology, which is the technology of the claimed invention.

Respectfully submitted,
Intel Corporation

Dated: August 4, 2008

/Molly A. McCall/ Reg. No. 46,126
Molly A. McCall
Intel Corporation
c/o Intellevate, LLC
P.O. Box 52050
Minneapolis, MN 55402